

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Infotehnoloogia eriala

**Heiki Pärn**

# **Graafiülesannete lahendamise keskkond**

Bakalaureusetöö (6 EAP)

Juhendaja: dots. R. Prank

Autor: ..... “.....” mai 2013

Juhendaja: ..... “.....” mai 2013

Lubada kaitsmisele

Professor: ..... “.....” mai 2013

TARTU 2013

# Sisukord

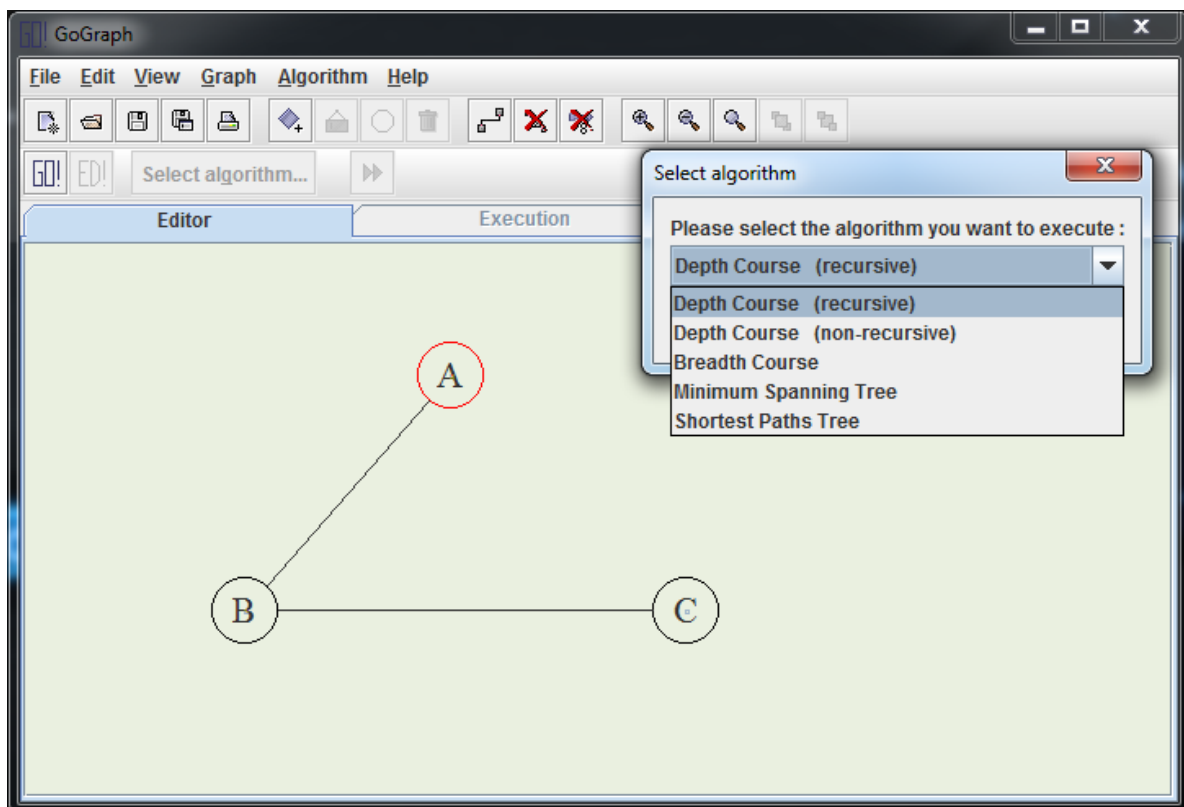
Sissejuhatus .....	3
1. Probleemi taust .....	5
1.1 Ainest.....	5
1.2 Graafiülesanded .....	5
2. Loodud keskkond .....	6
2.1 Eesmärk .....	6
2.2 Kasutajaliides .....	7
2.2.1 Graafi joonistamise paneel .....	7
2.2.2 Maatriksi konstrueerimise paneel .....	8
2.2.3 Tipuastmejärjendi konstrueerimise paneel .....	9
2.3 Realiseeritud ülesandetüübid .....	10
2.3.1 Naabrusmaatriksile vastava graafi joonistamine .....	11
2.3.2 Graafile vastava naabrusmaatriksi konstrueerimine .....	13
2.3.3 Täisgraafi joonistamine .....	14
2.3.4 Etteantud graafi täiendgraafi joonistamine .....	15
2.3.5 Graafile vastava tipuastmejärjendi konstrueerimine .....	17
2.3.6 Tipuastmejärjendile vastava graafi joonistamine .....	18
2.4 Graafiülesannete koostamise programm .....	20
2.4.1 Ülesandekogude loomine ja muutmine .....	20
2.4.2 Lahendusfailide kontrollimine.....	23
2.5 Graafiülesannete lahendamise programm .....	24
3. Realisatsioon.....	26
3.1 Kontrollsumma .....	26
3.2 Erdős-Gallai teoreem .....	26
Kokkuvõte .....	28
Summary.....	29
Viited .....	30
Lisad .....	31

# Sissejuhatus

Tartu Ülikooli aines „Diskreetse matemaatika elemendid“ (MTAT.05.109) kasutatakse mitmesuguseid õpiprogramme. Nende abil on lihtsam antud teemasid omandada ning teadmistekontrollle läbi viia.

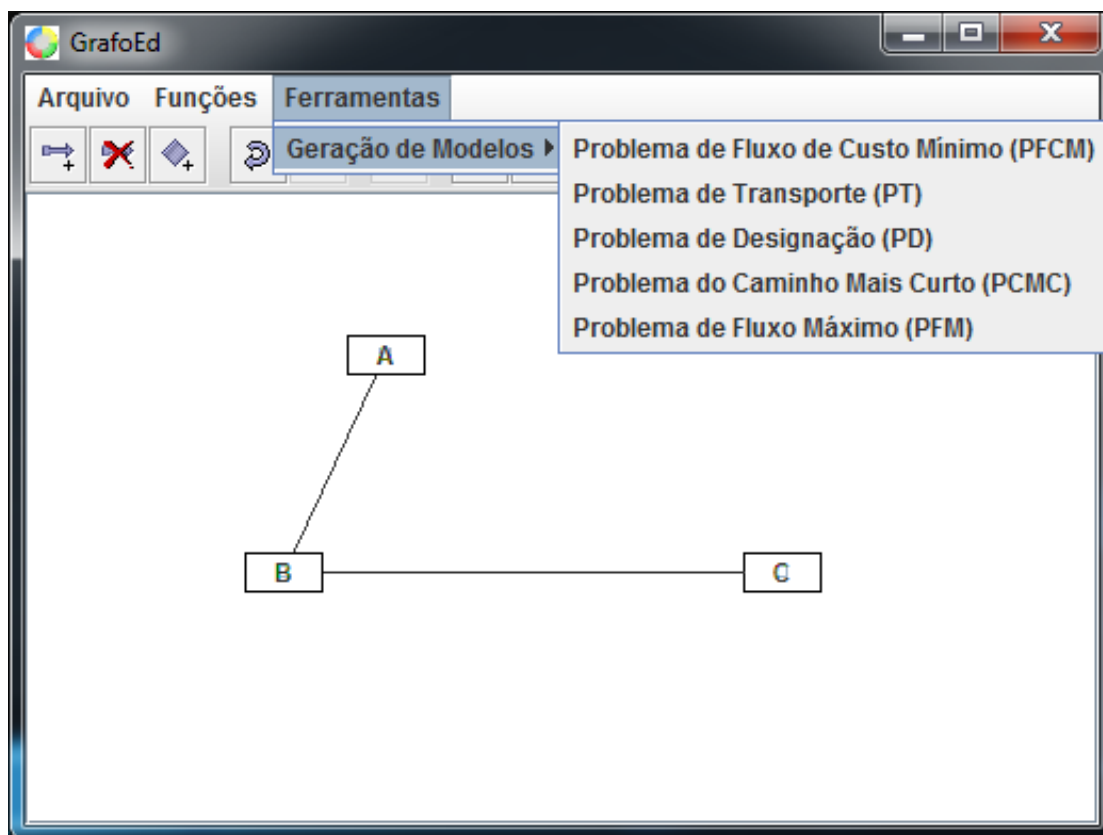
Üheks aines käsitletavatest teemadest on graafiteooria, mille jaoks pole seni õpiprogrammi kasutusele võetud. Internetist võib leida graafiteooriaga seotud programme. Järgnevalt on tutvustatud kaht vabavaralist java-rakendust: „GoGraph [1]“ ja „Graph Editor [2]“.

GoGraph abil on võimalik graafe joonistada ning nende peal erinevaid algoritme proovida. Programmis saab valida algustipu (*initial vertex*), millest lähtudes saab graafi läbi käia valitud algoritmi abil. Järgneval pildil on näidatud erinevad läbimisalgoritmid.



Joonis 1. GoGraph, tipp „A“ on valitud algustipuks.

Graph Editor-i ei olnud võimalik keelebarjääri tõttu väga põhjalikult testida, kuid tundub, et selle abil on samuti võimalik graafe erinevate algoritmide abil läbida.



Joonis 2. Graph Editor (GrafoEd)

Kuigi nende programmide abil on võimalik graafe joonistada, puudub neis võimalus luua ja lahendada ülesandeid. Selle bakalaureusetöö eesmärgiks oligi graafiülesannete lahendamise keskkonna prototüübi loomine, mille hulka kuuluvad graafiülesannete lahendamise programm ja graafiülesannete koostamise programm. Käesolevas töös kirjeldatakse nende kasutamist ja tuuakse välja nende erinevad võimalused.

Esimeses peatükis on kirjutatud ainek „Diskreetse matemaatika elemendid“ ja erinevatest võimalikest graafiteooria ülesandetüüpidest. Teises peatükis on kirjeldatud loodud keskkonna kasutajaliidest, realiseeritud ülesandetüüpe ja keskkonda kuuluvate koostamis- ja lahendusprogrammide kasutamist. Kolmandas peatükis on lähemalt seletatud kahte aspekti keskkonna realiseerimisest.

# 1. Probleemi taust

## 1.1 Ainest

Diskreetse matemaatika elemendid on kohustuslik aine infotehnoloogiat, informaatikat ja arvutitehnikat õppivatele tudengitele. See on esimese aasta ainetest üks raskemaid, kuid annab edaspidiseks vajalikud alustadmised.

Selle aine praktikumides on hetkel kasutusel kaks õpiprogrammi. Ühe abil saab lahendada tõeväärtustabelitel põhinevaid ülesandetüüpe ning teise abil teha lausearvutusevalemite teisendusi. Need programmid lihtsustavad vastavate teemade õpetamist.

## 1.2 Graafiülesanded

Graafiteooria õppimisel on kasulik lahendada ülesandeid. Raamatust [3] „Diskreetse matemaatika elemendid“ võib leida ja tuletada mitmesuguseid erinevat tüüpi ülesandeid. Nende hulka kuuluvad:

- naabusmaatriksile vastava graafi joonistamine
- graafile vastava naabusmaatriksi konstrueerimine
- täisgraafi joonistamine
- etteantud graafi täiendgraafi joonistamine
- graafile vastava tipuastmejärjendi konstrueerimine
- tipuastmejärjendile vastava graafi joonistamine
- intsidentsusmaatriksile vastava graafi joonistamine
- graafile vastava intsidentsusmaatriksi konstrueerimine
- kõikide  $n$ -tipuliste graafide joonistamine
- graafide isomorfisuse kontrollimine
- regulaarsete graafide joonistamine jne.

Loodud programmi valiti nendest kuus esimest ülesandetüüpi. Pikemalt on nende kohta kirjutatud jaotises 2.3.

## 2. Loodud keskkond

### 2.1 Eesmärk

Bakalaureusetöö eesmärgiks oli luua graafiülesannete lahendamise keskkonna algvariant. Sellesse kuuluvad ülesannete koostamise programm ja ülesannete lahendamise programm. Keskkond on kirjutatud programmeerimiskeeles Java. Koostamisprogrammi abil on võimalik luua ülesandekogusid, mis koosnevad kuut tüüpi ülesannetest:

1. naabrusmaatriksile vastava graafi joonistamine,
2. graafile vastava naabrusmaatriksi konstrueerimine,
3. täisgraafi joonistamine,
4. etteantud graafi täiendgraafi joonistamine,
5. graafile vastava tipuastmejärjendi konstrueerimine,
6. tipuastmejärjendile vastava graafi joonistamine.

Hetkel on toetatud ainult suunamata graafid, kuid tulevikus on võimalik juurde lisada ka suunatud graafid ja lisaks ka teisi ülesandetüüpe.

Lahendamisprogrammi abil saab luua lahendusfaile, andes ette ülesandekogu. Lahendusfaili kopeeritakse vastavast ülesandekogust kõik ülesanded. Avades lahendusfaili, saab lahendada selles olevaid ülesandeid. Lahenduste õigsust kontrollitakse programmi poolt ja tulemused salvestatakse samasse lahendusfaili. Hiljem saab lahendusfaili ülevaatlikult kontrollida koostamisprogrammi abil.

Järgnevates jaotistes on kirjeldatud selle keskkonna kasutajaliidest, erinevaid ülesandetüüpe ning koostamis- ja lahendamisprogrammi võimalusi.

## **2.2 Kasutajaliides**

Sõltuvalt ülesande tüübist koosneb kasutajaliides erinevatest elementidest. Näiteks naabrusmaatriksi järgi graafi joonistades on vaja paneeli graafi jaoks ja teist naabrusmaatriksi jaoks. Samasid paneele oleks vaja ka vastupidises ülesandes. Graafi joonistamise paneele läheb vaja ka täisgraafi ja täiendgraafi ülesandetüüpides. Tipuastmejärjendi järgi graafi joonistamisel või sellele vastupidises ülesandes läheb veel lisaks vaja paneeli tipuastmejärjendi jaoks.

Neid paneele läheks vaja nii ülesande koostamisel kui ka ülesande lahendamisel. Seega on meil valitud ülesandetüüpide jaoks vaja graafi joonistamise paneeli, maatriksi konstrueerimise paneeli ja tipuastmejärjendi konstrueerimise paneeli.

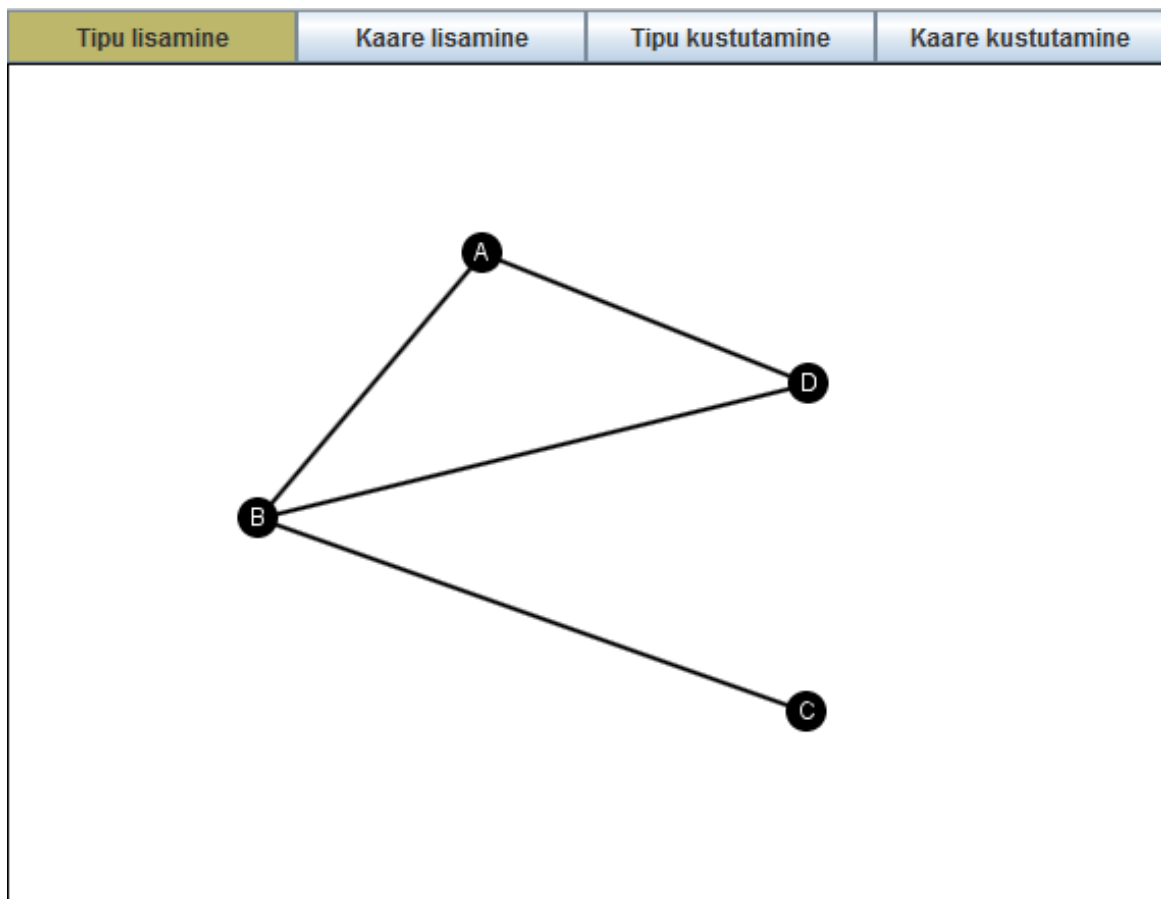
### **2.2.1 Graafi joonistamise paneel**

Graafi joonistamise paneelil saab luua graafi. Selle jaoks on võimalik lisada ja kustutada tippe. Tippe saab nimetada ja paneelil ringi liigutada. Lisaks sellele saab tippude vahele lisada ja sealt eemaldada kaari. Tipu kustutamisel eemaldatakse automaatselt ka kõik sellega ühendatud kaared.

Sellel paneelil on neli režiimi: tipu lisamine, tipu kustutamine, kaare lisamine ja kaare kustutamine. Tipu lisamise režiimis saab paneelile klõpsates tekitada uue tipu. Lisaks saab selles režiimis tippe liigutada ja ümber nimetada. Olemasolevale tipule klõpsates või pärast tipu lisamist on tipp aktiivne. Aktiivsele tipule saab nime anda, vajutades klaviatuuril soovitud täheklahvi. Tipu kustutamise režiimis saab tippe kustutada neile klõpsates. Paneeli loomisel määratakse maksimaalne tippude arv, mida sinna lisada saab.

Kaare lisamise režiimis saab tippude vahele lisada kaari. Selleks peab tegema klõpsu nende kahe tipu peal, mille vahele kaart lisada soovitakse. Kaari saab kustutada kaare kustutamise režiimis.

Liigutades hiirt tipu peale või kaare lähedale, muudavad nad värvi. See teeb programmi kasutamist mugavamaks.



Joonis 3. Graafi joonistamise paneel, antud paneelil on valitud tipu lisamise režiim.

### 2.2.2 Maatriksi konstrueerimise paneel

Maatriksi konstrueerimise paneelil saab luua naabusmaatriksit. Maatriks koosneb lahtritest, mis tähistavad tippude vahelisi kaari. Kaarelahtrite väärtus võib olla kas „0“ (tippude vahel pole kaart) või „1“ (tippude vahel on kaar). Paneelil on ka lahtrid rea ja veeru nimede jaoks. Maatriksi järku saab muuta.

Sellel paneelil on kaks nuppu: „suurenda järku“ ja „vähenda järku“. Neid vajutades suurendatakse või vähendatakse maatriksi järku ehk ridade ja veergude arvu. Paneeli loomisel määratakse maksimaalne võimalik maatriksi järk. Rea/veeru nimesid saab muuta lahtri peale klõpsates, mille tulemusena lahter aktiivseks läheb. Seejärel tuleb vajutada soovitud täheklahvi. Muutes maatriksis rea nime, muutub automaatselt ka vastav veerunimi ja



vastupidi. Kaarelahtrid on algselt tühjad aga neile peale klõpsates saab nende väärtust muuta.

Suurenda järku				Vähenda järku			
	A	B	C	D			
A	0	1	1	0			
B	1	0	1	0			
C	1	1	0	0			
D	0	0	0	0			

Joonis 4. Maatriksi konstrueerimise paneel

### 2.2.3 Tipuastmejärjendi konstrueerimise paneel

Viimaseks vajalikuks paneeliks on tipuastmejärjendi konstrueerimise paneel. Sellel saab luua tipuastmejärjendi, mille pikkust saab muuta. Järjendi elementideks on mitte-negatiivsed täisarvud.

Sarnaselt maatriksi konstrueerimise paneeliga on ka sellel paneelil „suurenda järku“ ja „vähenda järku“ nupud, mida vajutades suurendatakse või vähendatakse järjendi pikkust. Maksimaalne võimalik järjendi pikkus määratakse paneeli loomisel. Tipuastmejärjendiks on lahtrite rida. Lahtrid ise on tekstiväljad, millesse saab sisestada ainult numbreid.

Suurenda järku		Vähenda järku		
3	2	1	1	1

Joonis 5. Tipuastmejärjendi konstrueerimise paneel

## 2.3 Realiseeritud ülesandetüübid

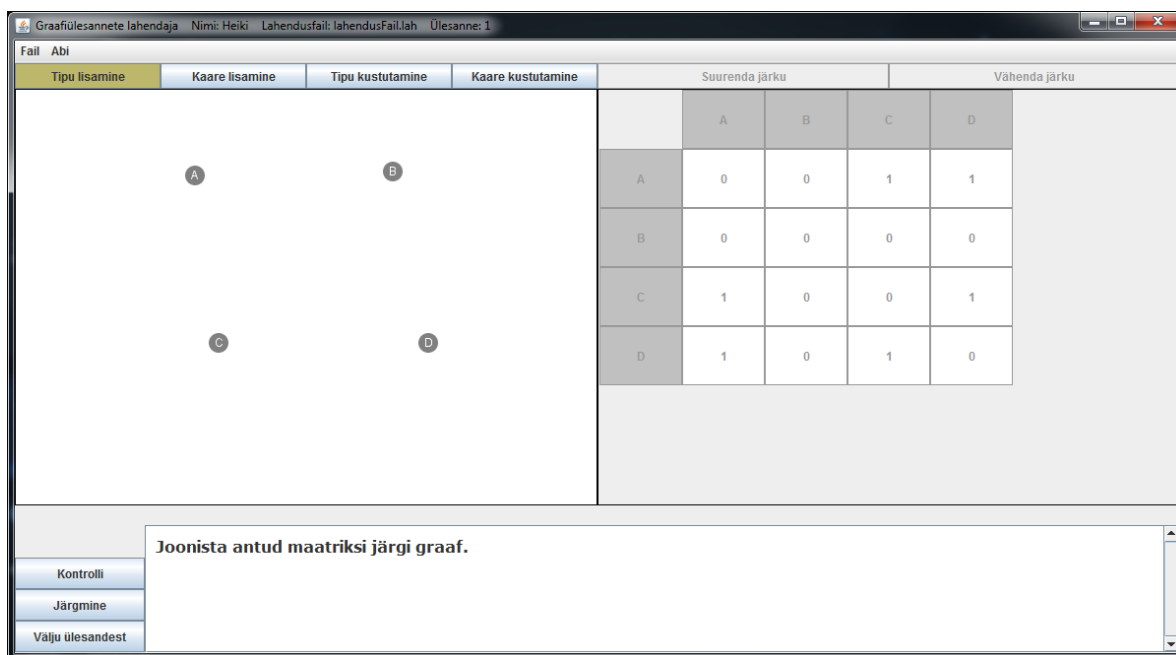
Praeguse seisuga on graafiülesannete keskkonnas olemas kuus erinevat ülesandetüüpi. Neis käsitletakse vastavalt tüübile graafi, naabrusmaatriksit ja tipuastmejärjendit.

Paljudes ülesandetüüpides võimaldab programm ülesande koostamisel selle vastuse osaliselt ette anda. Näiteks naabrusmaatriksi järgi graafi joonistamise ülesandes võib ette anda ainult maatriksi, aga lisaks maatriksile võib ette anda ka osa graafist. Sellisel juhul saab ülesande lahendaja etteantud graafi küll täiendada, kuid etteantud tippe ja kaari ta kustutada ei saa. Etteantud elemendid on teist värvi.

Järgnevalt on välja toodud realiseeritud ülesandetüüpide kirjeldused ja lisaks sellele ka nende kitsendused ülesande loomisel ja võimalikud vead lahendamisel. Kitsendused on vajalikud selleks, et ei saaks koostada vigaseid ülesandeid. Vead ülesande lahendamisel on jaotatud süntaksivigadeks ja sisulisteks vigadeks. Veatüüpide kohta on pikemalt kirjas jaotises 2.5.

### 2.3.1 Naabrusmaatriksile vastava graafi joonistamine

Selles ülesandetüübis on kasutusel maatriksi konstrueerimise paneel ja graafi joonistamise paneel. Maatriksi konstrueerimise paneelil on naabrusmaatriks, mida lahendaja muuta ei saa. Etteantud maatriksi järgi on vaja joonistada sellele vastav graaf graafi joonistamise paneelile. Graaf võib olla ka osaliselt ette antud.



Joonis 6. Naabrusmaatriksi järgi graafi joonistamine. Peale maatriksi on ette antud ka graafi tipud.

Kitsendused ülesande koostamisel:

- Maatriks ei tohi olla tühi.
- Maatriksi peadiagonaalil peab igas lahtris olema „0“.
- Maatriks peab olema sümmeetriline (kuna hetkel on toetatud ainult suunamata graafid).
- Maatriksi read ja veerud peavad olema nimetatud.
- Maatriksis ei tohi olla tühje lahtreid.
- Maatriksis ei tohi olla korduvaid rea/veeru nimesid.
- Graafi ette antud tipud peavad olema nimetatud.
- Graafi ette antud tippude arv ei saa olla suurem kui maatriksi järk.
- Graafil ei tohi olla korduvaid tipunimesid.
- Graafi ette antud tippude nimed ei tohi olla vastuolus maatriksi rea/veeru nimedega.
- Graafil ei tohi olla ette antud kaart, mis maatriksis on märkimata.

Süntaksivead lahendamisel:

- Mõni graafi tipp on nimetamata.
- Korduv tipunimi graafil.

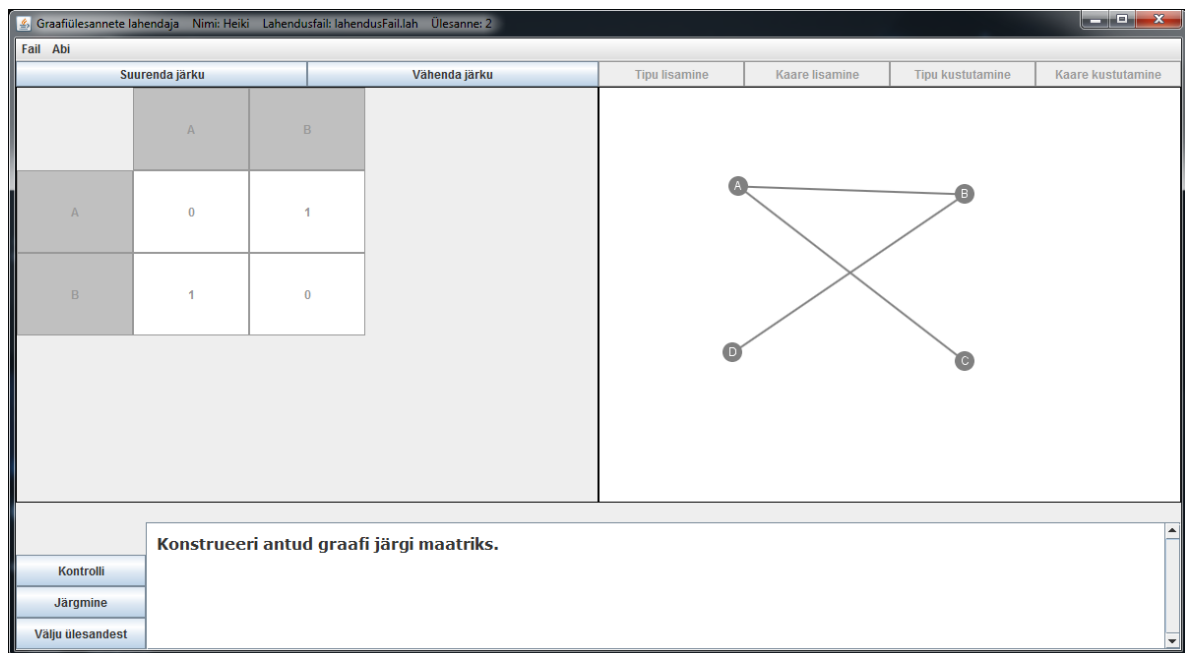
Sisulised vead lahendamisel:

- Vale tippude arv graafil.
- Graafi tippude nimed ei ühti maatriksi rea/veeru nimedega.
- Graafil on liigne kaar.
- Graafilt on puudu kaar.
- Kaared graafil on valesti (kui on nii liigseid kui ka puuduvaid).

### 2.3.2 Graafile vastava naabrusmaatriksi konstrueerimine

See ülesandetüüp on vastupidine eelmisele. Siin on samuti kasutusel maatriksi konstrueerimise paneel ja graafi joonistamise paneel. Ette on antud graaf, mida lahendaja muuta ei saa. Graafi järgi on vaja joonistada sellele vastav maatriks.

Maatriks võib olla osaliselt ette antud. Kõikidele etteantava maatriksi ridadele ja veergudele tuleb nimi anda. Maatriksi järk ei saa olla suurem, kui graafi tippude arv. Maatriksis võib olla nii väärtustatud (0 või 1) kui ka tühje lahtreid.



Joonis 7. Graafi järgi naabrusmaatriksi konstrueerimine. Ette on antud ka osa maatriksist.

Kitsendused ülesande koostamisel:

- Graaf ei tohi olla tühi.
- Graafi tipud peavad olema nimetatud.
- Graafil ei tohi olla korduvaid tipunimesid.
- Maatriksi ette antud read ja veerud peavad olema nimetatud.
- Maatriksi järk ei saa olla suurem kui graafi tippude arv.

- Maatriksis ei tohi olla korduvaid tipunimesid.
- Maatriksi peadiagonaalil peab igas mittetühjas lahtris olema „0“.
- Maatriks peab olema sümmeetriline.
- Maatriksis ei tohi „1“ olla elementide vahel, mis graafil on kaarega ühendamata.
- Maatriksi ette antud rea/veeru nimed ei tohi olla vastuolus graafi tipunimedega.

Süntaksivead lahendamisel:

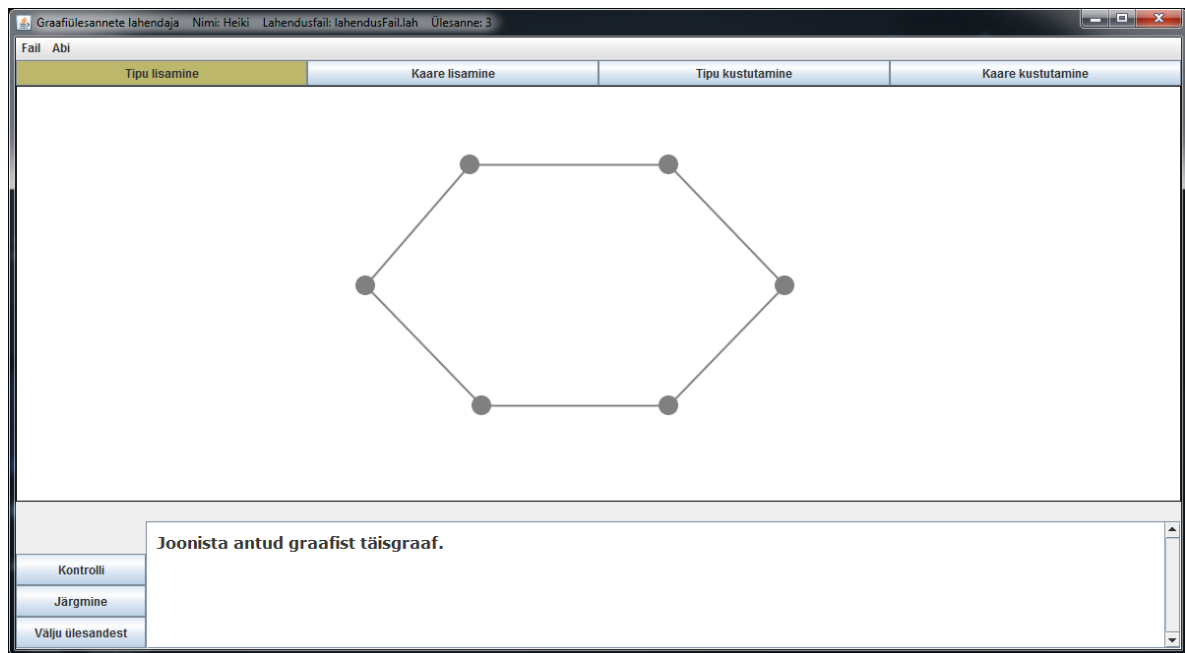
- Mõni maatriksi rida/veerg on nimetamata.
- Korduv rea/veeru nimi maatriksis.
- Mõni maatriksi lahter on tühi.

Sisulised vead lahendamisel:

- Vale maatriksi järk.
- Maatriksi rea/veeru nimed ei ühti graafi tipunimedega.
- Maatriksis on liigne „kaar“.
- Maatriksist on puudu „kaar“.
- Viga maatriksi lahtrites (kui on nii liigseid kui ka puuduvaid kaari).

### 2.3.3 Täisgraafi joonistamine

Selles ülesandetiübis on kasutusel ainult graafi joonistamise paneel. Ette on antud graaf, mille tippude hulka muuta ei saa. Samuti ei saa eemaldada juba olemasolevaid kaari. Graafile peab kaari lisama nii, et tekiks täisgraaf. Graafi tipud võib koostamisel nimetada või jätta nimetamata.



Joonis 8. Täisgraafi joonistamine.

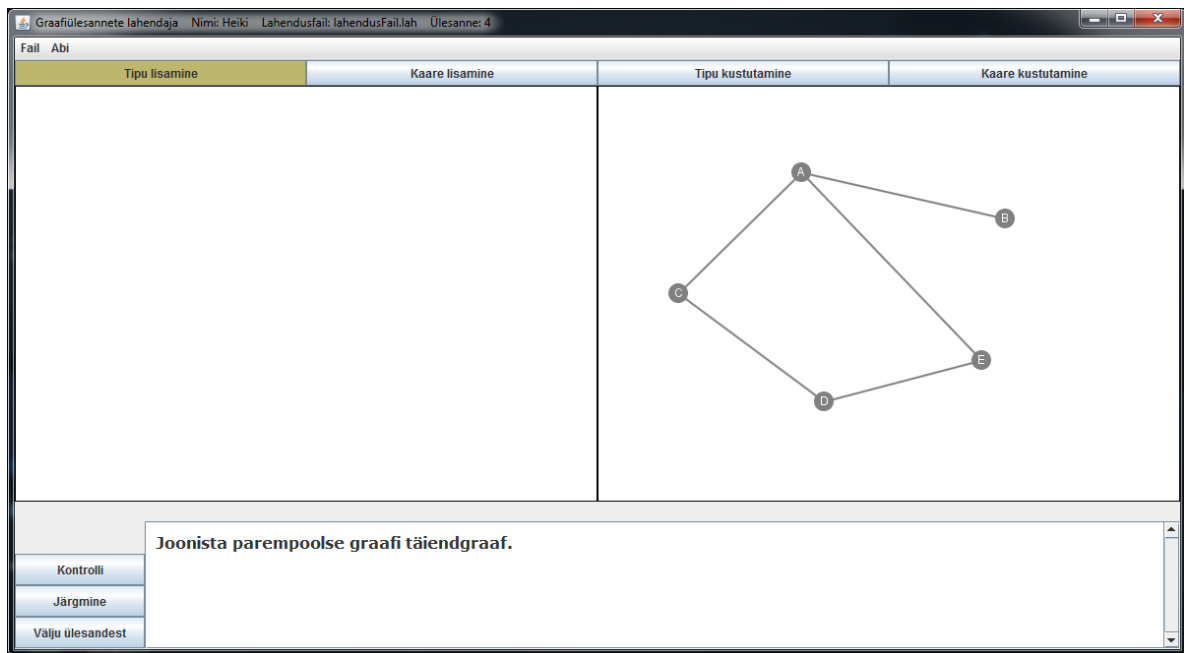
Kitsendused ülesande koostamisel:

- Graaf ei tohi olla tühi.
- Graafil ei tohi olla korduvaid tipunimesid.

Selle ülesandetüübi lahendamisel süntaksivead puuduvad ja sisulisi vigu ei eristata. Vale vastuse korral teatab programm „Vale vastus“.

### 2.3.4 Etteantud graafi täiendgraafi joonistamine

Siin on kasutusel kaks graafi joonistamise paneeli. Parempoolsel paneelil on ette antud graaf. Vasakpoolsele paneelile tuleb joonistada parempoolse täiendgraaf. Vasakpoolne graaf võib olla osaliselt ette antud.



Joonis 9. Täiendgraafi joonistamine.

Kitsendused ülesande koostamisel:

- Parempoolne (etteantav) graaf ei tohi olla tühi.
- Parempoolse graafi tipud peavad olema nimetatud.
- Parempoolse graafi tipunimed ei tohi korduda.
- Vasakpoolse graafi tipud peavad olema nimetatud.
- Vasakpoolse graafi tipunimed ei tohi korduda.
- Vasakpoolsel graafil ei tohi olla rohkem tippe kui parempoolsel.
- Graafide tipunimed ei tohi olla vastuolus.
- Graafidel ei tohi olla ühiseid kaari.

Süntaksivead lahendamisel:

- Mõni graafi tipp on nimetamata.
- Korduv tipunimi graafil.

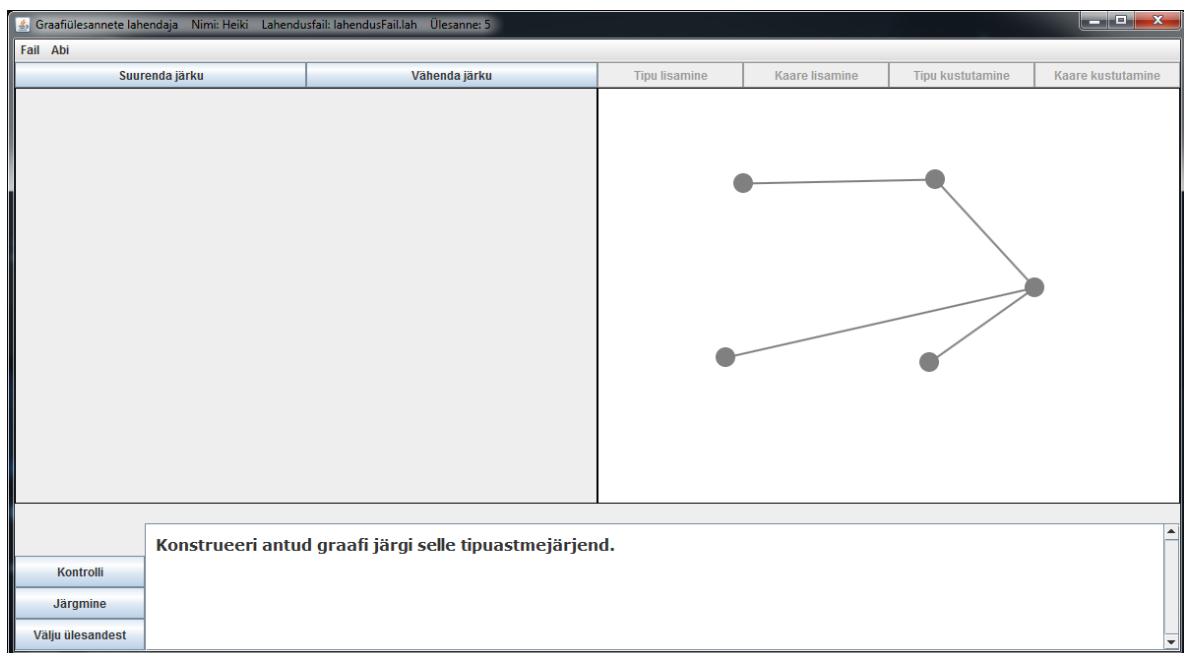


Sisulised vead lahendamisel:

- Vale tippude arv graafil.
- Graafide tippude nimed ei ühti.
- Graafi kaared on valesti.

### 2.3.5 Graafile vastava tipuastmejärjendi konstrueerimine

Selles ülesandetüübis on kasutusel graafi joonistamise paneel ja tipuastmejärjendi joonistamise paneel. Ette on antud graaf, mida muuta ei saa. Graafi tipud võib koostamisel nimetada või jätta nimetamata. Graafi järgi tuleb konstrueerida sellele vastav tipuastmejärjend. Siin ülesandes pole võimalik „vasakut poolt“ ehk tipuastmejärjendit ette anda.



Joonis 10. Graafi järgi tipuastmejärjendi konstrueerimine.

Kitsendused ülesande koostamisel:

- Graaf ei tohi olla tühi.
- Graafi tipunimed ei tohi korduda.

Süntaksivead lahendamisel:

- Mõni tipuastmejärjendi lahter on tühi.
- Tipuastmejärjend ei ole mittekasvav.

Sisulised vead lahendamisel:

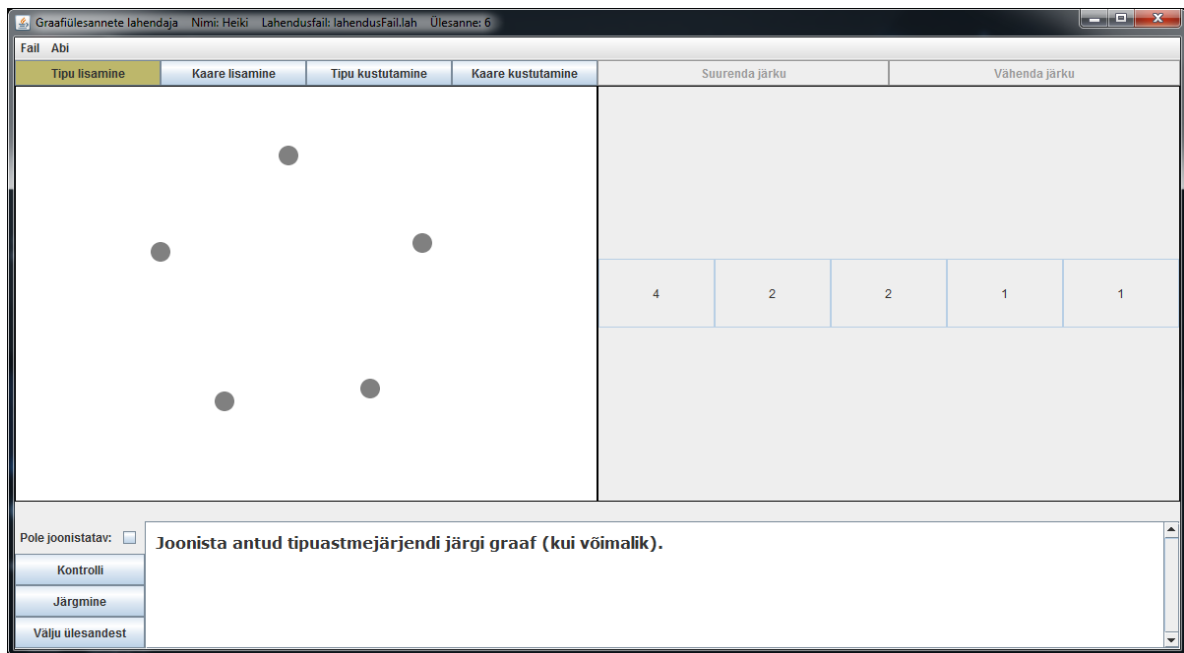
- Vale lahtrite arv tipuastmejärjendis.
- Viga tipuastmejärjendis.

### **2.3.6 Tipuastmejärjendile vastava graafi joonistamine**

See ülesandetüüp on vastupidine eelmisele. Ette on antud tipuastmejärjend, mida muuta ei saa. Järjendi järgi tuleb joonistada sellele vastav graaf. Siin ülesandes on võimalik graafi tippe, kuid mitte kaari, ette anda. Graafi tipud võib koostamisel nimetada või jätta nimetamata.

Selle ülesande teeb eriliseks asjaolu, et kui iga graafi järgi saab konstrueerida sellele vastava tipuastmejärjendi, siis iga tipuastmejärjendi järgi ei ole võimalik graafi joonistada. Sellisel juhul saab lahendamisel ära märkida „Pole joonistatav“ märkeruudu. See märkeruut on kasutusel ainult selles ülesandetüübis. Sellest, kuidas programm kontrollib, kas järjendi järgi saab graafi joonistada või mitte, on pikemalt kirjas jaotises 3.2.

Aines „Diskreetse matemaatika elemendid“ õpetatakse, et kui tipuastmejärjendi elementide summa ei ole paarisarv, siis ei leidu sellele järjendile vastavat graafi. See ei ole siiski ainus kriteerium. Juhul, kui etteantud tipuastmejärjendi elementide summa on paarisarv, aga selle põhjal ei saa ikkagi graafi joonistada, küsib koostamisprogramm ülesande salvestamisel kinnitust, kas ülesande koostaja ikka soovib seda ülesannet salvestada.



Joonis 11. Tipuastmejärjendi järgi graafi joonistamine. Ette on antud ka graafi tipud.

Kitsendused ülesande koostamisel:

- Tipuastmejärjend ei tohi olla tühi.
- Tipuastmejärjendis ei tohi olla tühju lahtreid.
- Tipuastmejärjend peab olema mittekasvav.
- Graafil ei tohi olla rohkem tippe kui on tipuastmejärjendis lahtreid.
- Graafil ei tohi olla korduvaid tipunimesid.

Ainus süntaksiviga selle ülesande lahendamisel on see, et juhul kui tipud nimetada (ei ole kohustuslik), siis ei tohi olla korduvaid tipunimesid.

Sisulised vead lahendamisel:

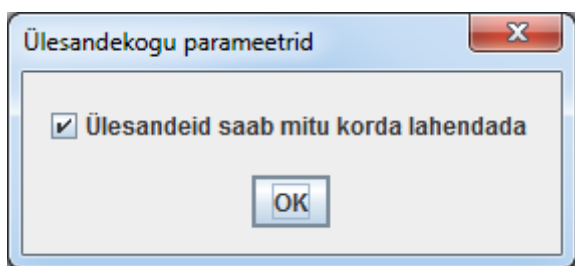
- On joonistatav (kui on märgitud kast „Pole joonistatav“, kuid tegelikult on võimalik graafi siiski joonistada).
- Vale tippude arv graafil.
- Joonistatud graafil on teistsugune tipuastmejärjend.

## 2.4 Graafiülesannete koostamise programm

Graafiülesannete lahendamise keskkond on jaotatud kaheks osaks. Esimene neist on „Graafiülesannete koostaja“. Selle programmi abil on võimalik luua ja muuta ülesandekogusid ning kontrollida lahendusfaile.

### 2.4.1 Ülesandekogude loomine ja muutmine

„Graafiülesannete koostaja“ programmi abil saab luua ja muuta ülesandekogusid, mida lahendamisprogrammi abil hiljem lahendada saab. Uue ülesandekogu loomiseks tuleb pärast programmi käivitamist menüüst valida „Loo ülesandekogu“. Seejärel tuleb valida koht, kuhu fail salvestada, ja loodava ülesandekogu parameetrid.

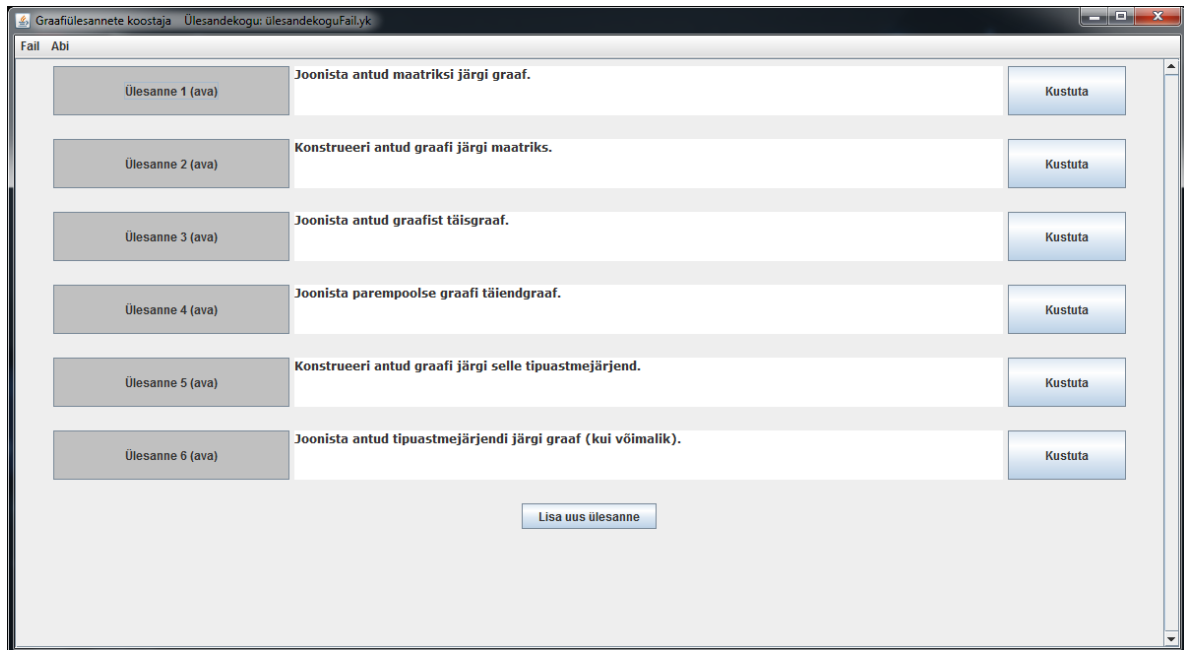


Joonis 12. Ülesandekogu parameetrid.

Parameetreid on hetkel ainult üks: kas ülesandeid saab lahendada mitu korda või ainult ühe korra. Kui lubada ainult ühekordne lahendamine, on lubatud ainult üks lahenduskord. Kui seda piirangut pole, võib ülesandeid lahendada niikaua, kuni saadakse õige tulemus, hoolimata lahenduskorrast. Sellisel juhul võib neid ka pärast õige tulemuse saamist uuesti lahendada, ja kuigi tehtavad vead märgitakse lahendusfaili üles, ei mõjuta need enam ülesande staatust. Ülesande staatuse kohta on pikemalt kirjas jaotises 2.5.

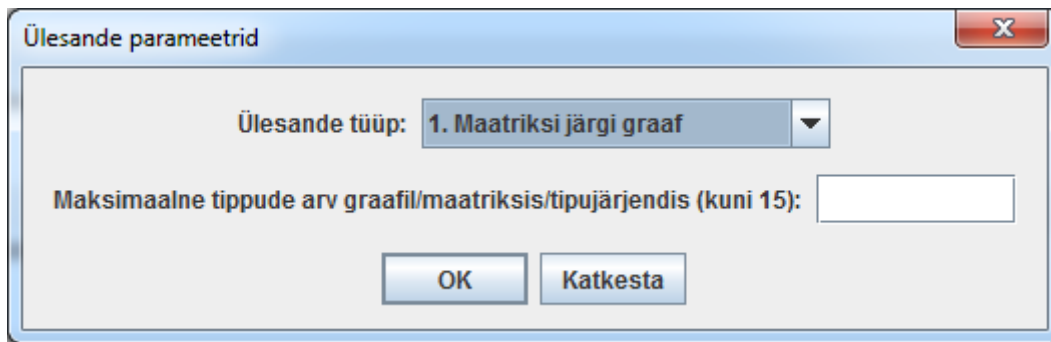
Kui ülesandekogu on loodud, siis see avatakse automaatselt. Lisaks on võimalik avada muutmiseks juba olemasolevaid ülesandekogusid valides menüüst „Ava ülesandekogu“ ja näidates, kus asub ülesandekogu fail. Pärast ülesandekogu avamist kuvatakse programmi üleval ääres ülesandekogu faili nimi.

Kui ülesandekogu on avatud, siis on võimalik sinna lisada ja sealt eemaldada ülesandeid. Lisatud ülesande peale klõpsates saab selle algandmeid muuta. Uue ülesande lisamiseks on olemas nupp „Lisa uus ülesanne“ ja kustutamiseks on nupp „Kustuta“.



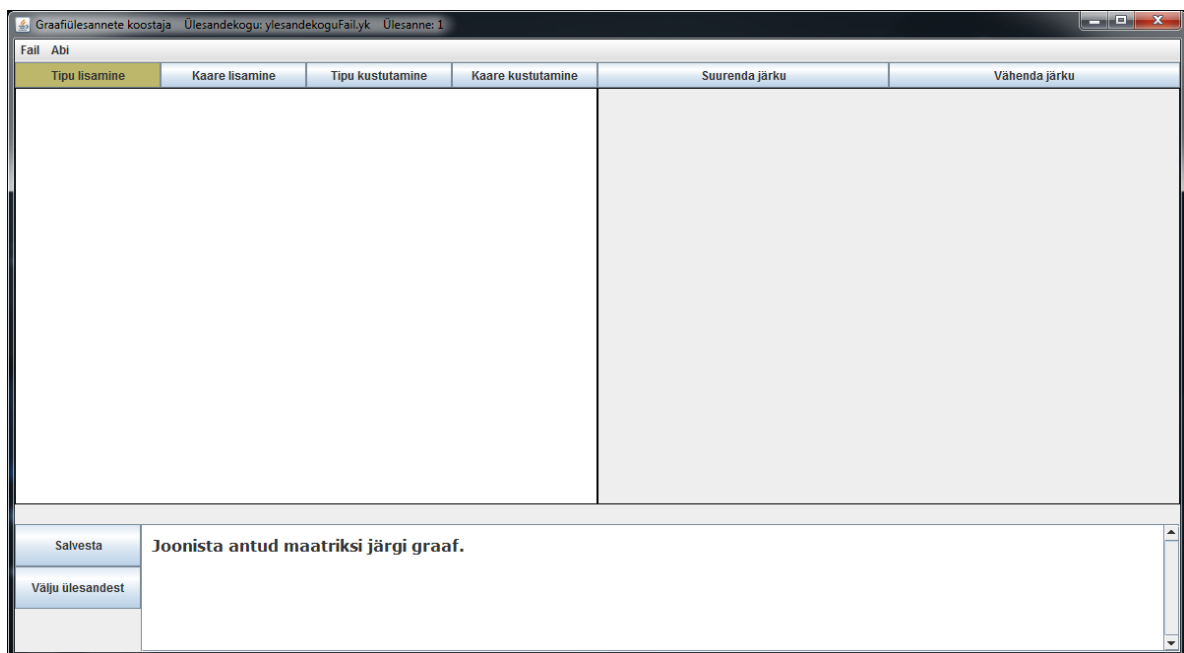
Joonis 13. Ülesannete nimekirja kuvamine koostamisprogrammis.

Pärast ülesande lisamise nupu vajutamist kuvatakse aken, milles saab määrata ülesande parameetrid. Hetkel on nendeks ülesande tüüp ja vastava ülesande maksimaalne tippude arv. Maksimaalne tippude arv tähendab seda, mitu tippu saab selle ülesande koostamisel maksimaalselt graafile joonistada või mitu elementi saab maksimaalselt olla maatriksis või tipuastmejärjendis. Maksimaalne tippude arv võib olla suurem, kui on õiges vastuses.



Joonis 14. Ülesande parameetrid.

Pärast ülesande parameetrite seadmist läheb programm ülesande loomise paneelile. Seal saab ette anda algandmed. Ülesande tekst on vastavalt tüübile algselt ette antud, aga seda on võimalik muuta. Ülesande koostamisel kuvatakse akna ülemises ääres lisaks ülesandekogu faili nimele ka ülesande number.



Joonis 15. "Maatriksi järgi graaf" tüüpi ülesande loomine. Graafi joonistamise paneel, maatriksi konstrueerimise paneel ja tekstiala on kõik vabalt muudetavad.

Pärast soovitud muudatuste tegemist saab ülesande salvestada vajutades „Salvesta“ nupule. Koostamisprogramm kontrollib, kas ülesanne on korrektne. Kui ei ole, antakse sellest teada. Korrekse ülesande puhul salvestatakse ülesande andmed ülesandekogu faili.

Ülesandest saab väljuda vajutades „Välju ülesandest“ nuppu, mille tulemusena kuvab programm taas ülesannete nimekirja. Kui seda nuppu vajutades pole ühtegi korda ülesannet salvestatud, küsib programm üle, kas ülesandest ikka soovitakse väljuda.

### **2.4.2 Lahendusfailide kontrollimine**

Lisaks ülesannete loomisele saab koostamisprogrammis kontrollida ka lahendusfaile. Ülesannete lahendamisel salvestatakse lahendusfaili lahenduskorrad ja tehtud vead. Osalist lahendust praeguses realisatsioonis ei salvestata.

Lahendusfaili kontrollimiseks tuleb menüüst valida „Kontrolli lahendusfaili“. Seejärel tuleb valida lahendusfaili asukoht. Pärast faili avamist kuvatakse paneeli üleval ääres lahendaja nimi ja see, kas kontrollsumma on korrektne. Kontrollsumma kohta on lähemalt kirjas jaotises 3.1.

Nende all on tabel, kus on näha kõikide ülesannete lahenduskorrad ja lahenduskordade ajal tehtud vead. Iga ülesandesse sisenemine läheb lahenduskorrana arvesse. Nendel lahenduskordadel, mille jooksul ülesanne edukalt lahendati, on taga „(+““. Kui ühe lahenduskorra ajal tehakse sama viga mitu korda, märgitakse alates teisest korrast vea taha sulgudesse, mitmes kord see viga esineb.

Graafiülesannete koostaja		
Fail Abi		
Nimi: Heiki		Lahendusfail: lahendusfail.lah
Ülesanne 1 - õigesti	Lahendus 1 (+)	Viga: Graafilt on puudu kaar!
		Viga: Graafilt on puudu kaar! (2)
		Viga: Vale tippude arv graafil!
Ülesanne 2 - õigesti	Lahendus 1 (+)	Viga: Maatriksilt on puudu kaar!
Ülesanne 3 - valesti	Lahendus 1	-
	Lahendus 2	Viga: Vale vastus!
		Viga: Vale vastus! (2)
Ülesanne 4 - lahendamata	-	-
Ülesanne 5 - lahendamata	-	-
Ülesanne 6 - lahendamata	-	-

Joonis 16. Lahendusfaili kontrollimine.

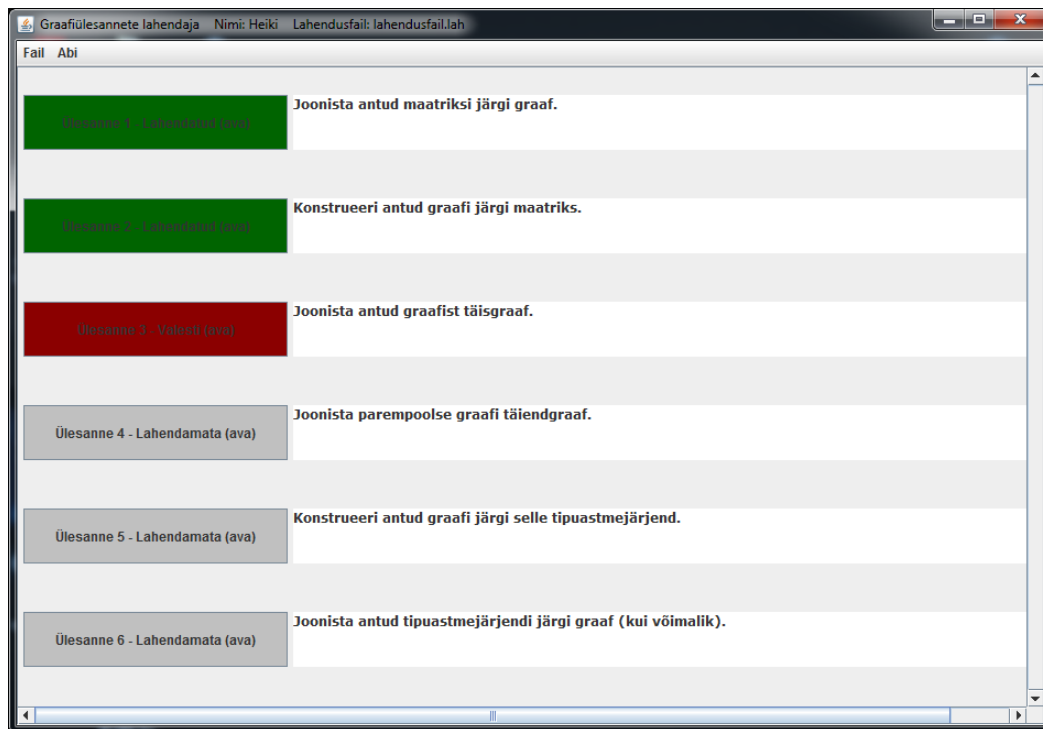
## 2.5 Graafiülesannete lahendamise programm

„Graafiülesannete lahendaja“ on teine osa graafiülesannete keskkonnast.

Lahendamisprogrammi abil on võimalik luua uus lahendusfail valides menüüst „Ava ülesandekogu“. Lahendusfaili saab luua, andes ette ülesandekogu. Esiteks tuleb valida ülesandekogu faili asukoht ja seejärel salvestada uus lahendusfail. Olemasolevaid lahendusfaile saab avada, valides menüüst „Ava lahendusfail“.

Pärast lahendusfaili avamist kuvatakse lahendusfailis olevate ülesannete nimekiri. Ülesannetel on staatus, milleks saab olla kas „lahendamata“, „õigesti lahendatud“ või „valesti lahendatud“. Lahendamata ülesanded on kuvatud halli, õigesti lahendatud ülesanded rohelise ja valesti lahendatud ülesanded punase värviga. Programmi üleval ääres kuvatakse lahendaja nimi ja lahendusfaili nimi.





Joonis 17. Ülesannete nimekirja kuvamine. Esimesed kaks ülesannet on märgitud rohelise kastiga, mis tähendab, et nad on õigesti lahendatud. Kolmas ülesanne on punane ehk valesti lahendatud. Ülesanded 4-6 on hallid ehk lahendamata.

Ülesandel klõpsates saab selle avada. Seejärel saab avatud ülesannet lahendada. Kuna osalist lahendust ei salvestata, ei saa ülesande lahendamist jätkata varem pooleli olnud kohast, ja iga kord kuvatakse ülesanne uuesti algandmete põhjal.

Üleval akna ääres kuvatakse pärast ülesande avamist lisaks selle ülesande number. Pärast lahendamist saab lahenduse õigsust kontrollida, vajutades „Kontrolli“ nuppu. Programm teostab kontrolli ja teatab, kas vastus oli õige või tehti mingi viga. Pärast ülesande lahendamist muudetakse lahendusfailis ülesande staatust. Tehtavad vead märgib programm lahendusfaili üles.

Eristatakse kahte tüüpi vigu: süntaksivead ja sisulised vead. Süntaksivigade alla kuuluvad näiteks tippude nimetamata jätmine (kui see vastavas ülesandetüübis nõutud on) või maatriksi lahtrite tühjaks jätmine. Süntaksivea tegemisel saab lahendamist jätkata ja nende tegemine ei mõjuta ülesande staatust. Sisuliste vigade alla kuuluvad näiteks tippude valesti nimetamine või vale kaarte arv. Nende tegemisel muudetakse ülesande staatust.

Ülesannetes saab igal ajal vajutada „Välju ülesandest“ nuppu, mille tulemusena kuvatakse taas kõik ülesanded. Ülesandest väljudes ei muudeta ülesande staatust. Lisaks saab „Järgmine“ nuppu vajutades asuda kohe järgmise ülesande kallale. Kui seda nuppu viimase ülesande juures vajutada, kuvatakse jällegi kõik ülesanded.

## **3. Realisatsioon**

### **3.1 Kontrollsumma**

Nii ülesandekogudes kui ka lahendusfailides on kasutusel kontrollsumma. Selle peamiseks eesmärgiks on faili tervikluse tagamine. Kontrollsumma arvutamiseks kasutatakse „MD5“ räsifunktsiooni.

Faili avades ning enne igat faili muutvat tegevust arvutatakse faili kontrollsumma alates teisest reast ja võrreldakse seda esimesele reale kirjutatud väärtusega. Juhul, kui väärtused ei ole võrdsed, sulgeb programm faili ja annab teada, et fail on vigane. Kui väärtused on võrdsed, teeb programm faili vajalikud muudatused ning arvutab uue kontrollsumma, mis kirjutatakse esimesele reale.

### **3.2 Erdős-Gallai teoreem**

Ülesandetüübis „Tipuastmejärjendi järgi graafi konstrueerimine“ on võimalik ülesande koostamisel ette anda järjend, mille järgi ei saa graafi joonistada. Selle ülesandetüübi puhul on lahendamisel võimalik valida märkeruut „Pole joonistatav“. Algselt oli probleemiks see, millise algoritmi abil kontrollida, kas graafi saab joonistada või mitte. Õnneks leidis sellele suhteliselt lihtne ja efektiivne lahendus. See probleem sai lahendatud kasutades Erdős-Gallai teoreemi [4]:

Tipuastmejärjendile ( $d_1 \geq d_2 \geq \dots \geq d_n$ ) vastab mingi harilik graaf, kui kõigi järjendi elementide summa on paarisarv ja kehtib:

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{j=k+1}^n \min(d_j, k) \quad k = 1, \dots, n$$

Näiteks kui võtta tipuastmejärjend  $\{2, 1, 1, 1\}$ , siis selle järjendi puhul on lihtne näha, et graafi ei saa joonistada, kuna järjendi elementide summa ei ole paarisarv.

Kui aga võtta tipuastmejärjendiks  $\{4, 3, 1, 1, 1\}$ , siis elementide summa on ikkagi paarisarv, kuid selle järgi pole võimalik graafi joonistada. Antud teoreemi abil saab näidata, et sellele järjendile ei vasta harilikku graafi, kuna teoreemi tingimus ei kehti juhul, kui  $k$  on võrdne 2-ga.

## Kokkuvõte

Aines „Diskreetse matemaatika elemendid“ (MTAT.05.109) kasutatakse õpetamisel mitme-suguseid õpiprogramme. Hetkel ei ole kasutusel programmi graafiteooria õpetamiseks. Käesoleva bakalaureusetöö eesmärgiks oligi luua graafiülesannete lahendamise keskkonna prototüüp.

Keskkond koosneb kahest programmist: „graafiülesannete koostaja“ ja „graafiülesannete lahendaja“. Koostamisprogrammi abil saab koostada ülesandekogusid. Lahendusprogrammis saab kogude põhjal luua lahendusfaile ja nendes ülesandeid lahendada. Lahendusfaile saab koostamisprogrammi abil ülevaatlikult kontrollida.

Hetkel on keskkonnas võimalik luua ja lahendada kuut tüüpi ülesandeid:

1. naabusmaatriksile vastava graafi joonistamine,
2. graafile vastava naabusmaatriksi konstrueerimine,
3. täisgraafi joonistamine,
4. etteantud graafi täiendgraafi joonistamine,
5. tipuastmejärjendile vastava graafi joonistamine,
6. graafile vastava tipuastmejärjendi konstrueerimine.

Nii ülesannete loomisel kui lahendamisel läheb vaja kolme paneeli: graafi joonistamise paneel, maatriksi konstrueerimise paneel ja tipuastmejärjendi konstrueerimise paneel.

Ülesande loomisel on kitsendused selle jaoks, et ei saaks luua vigaseid ülesandeid. Lahendamisel kontrollitakse erinevat tüüpi vigade esinemist.

Üheks huvitavamaks probleemiks oli see, kuidas programmis kontrollida, kas järjendi põhjal saab graafi joonistada või mitte. Lahenduseks oli Erdős-Gallai teoreemi kasutamine.

Tulevikus on võimalik keskkonda lisada suunatud graafide tugi ning lisaks sellele ka erinevaid ülesandetüüpe.

# Summary

## Graph theory exercise environment

### Bachelor's thesis

### Heiki Pärn

In Elements of Discrete Mathematics (MTAT.05.109) course various programs are used for teaching. Currently there is no program for creating and solving graph theory exercises. The purpose of this Bachelor's thesis was to create a prototype for graph theory exercise environment. Only undirected graphs are supported in the prototype and there are six types of exercises:

1. drawing a graph based on a given adjacency matrix,
2. constructing an adjacency matrix based on a given graph,
3. drawing a complete graph,
4. drawing a complement graph of a given graph,
5. constructing a degree sequence based on a given graph,
6. drawing a graph based on a given degree sequence.

Three panels are used in different combinations for these six exercises: graph panel, matrix panel and degree sequence panel. They are used to create and solve exercises.

When creating an exercise there are constraints to prevent incorrect or empty exercises. When solving the exercises the program checks for different types of errors. All errors are recorded in the file and can be reviewed later.

One of the problems encountered was that a suitable algorithm had to be found to check whether a degree sequence is graphical or not. This problem was solved using the Erdős-Gallai theorem.

In the future it is possible to add support for directed graphs. New exercise types can also be added.

## Viited

1. GoGraph. <http://sourceforge.net/projects/gograph/>
2. Graph Editor. <http://grapheditor.sourceforge.net/>
3. Reimo Palm. Diskreetse matemaatika elemendid. Tartu 2003.
4. M. Aigner, E. Triesch. Realizability and uniqueness in graphs. Discrete Mathematics. 136, 1–3, 3–20. 1994.  
<http://www.sciencedirect.com/science/article/pii/0012365X9400104Q>

## **Lisad**

Lisa 1: CD plaat koos graafiülesannete keskkonna programmide ja lähtekoodiga.

**Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Heiki Pärn (sünnikuupäev: 11.05.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Graafiülesannete lahendamise keskkond“, mille juhendaja on Rein Prank
  - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **13.05.2013**